Optimal Datapath Generation Using ACTgen

Logic systems consist of two basic elements: control logic an datapath logic. Control logic consists of state machines and other miscellaneous logic. Datapath logic consists of functions like counters, arithmetics, and memory. As device complexity increases, datapath logic begins to dominate as an overall percentage of the design. Studies by ASIC companies have shown that datapath logic represents more than 70% of the average design at densities above 20,000 gates.

With this increase in datapath logic, it has become imperative that optimized datapath logic be readily available to maximize silicon value. This is true whether the design is synthesis based, schematic based, or some mix of the two. This Application Note describes Actel's support of datapath logic functions, shows how they are used interactively in schematic capture systems, and how they operate transparently when working with synthesis tools.

The Datapath Synthesis Engine

Actel's ACTgen is the central resource for building datapath logic functions. ACTgen is built on a datapath synthesis engine with hand-optimized algorithms that have been painstakingly designed to maximize performance and to minimize logic costs. When these two goals are in conflict, ACTgen provides a variety of performance/cost options from which to choose. Each macro has also been optimized over a variety of widths and with various control options. A good example is a basic counter that has several performance/cost variations and a variety of control options. Each variation can be configured from 2 through 32 bits in width. Each variation has optional load, count enable, and asynchronous clear options. Each control input can be active high or active low. And finally, a terminal count output is available if required. Therefore, for a single simple counter function, ACTgen can build thousands of variations to cover almost any need required by an application. Table 1 illustrates the performance/cost trade-offs for three counter variations.

Table 1 • *Example of Performance/Cost Variations for a 24-bit Counter Macro in an ACT 3, -3 Speed Device*

Variation Type	Area	Performance MHz
Compact	39	30
Register Look-ahead	52	110
Pre-scaled	72	232

In addition to counters, the following datapath elements are available through ACTgen. Similar to counters, many speed/area variations and control options are available for each macro type.

- Arithmetic - adders, subtracters, accumulators, and multipliers
- Register functions - storage registers, storage latches, and shift registers
- Comparator functions - equality, magnitude, and constant decodes (fixed equality)
- Memory - RAM and FIFO
- Multiplexers
- Decoders
- Other basic logic functions

An engine is only as good as the mechanism that delivers its power and ACTgen has two options: The first option is through automatic inferencing used with commercial synthesis tools; the second option is to interactively work with the ACTgen Macro Builder and manually instantiate macros. The following sections further describe these mechanisms.

Automatic and Transparent Use by Synthesis Tools

Synthesis has several goals including vendor independence and improved productivity. Classical synthesis is superb at implementing random and control logic; however, generic algorithms have often failed to deliver either adequate performance or utilization for datapath logic elements because of their specialized natures. The situation is further complicated by complex FPGA architectures. This problem can be corrected transparently by incorporating the specialized algorithms of ACTgen's datapath synthesis engine directly into the synthesis process to realize hand optimized results. This process is automatic and preserves vendor independence.

There are two ways that synthesis can infer functions built by this engine. One way is for the synthesis tools to call ACTgen directly using either Actel's proprietary interface or using special libraries similar to Synopsys' DesignWare. The second method is for synthesis vendors to build custom algorithms using ACTgen as a benchmark for performance and utilization.

The second method eliminates file interaction which can have a positive impact on execution times. Table 2 defines the method used by various synthesis compilers.

Table 2 • *ACTgen's Access Methods for Synthesis*

Synthesis Compiler	Method
ACTmap (Actel provided compiler)	Proprietary Interface
Exemplar Galileo/Leonardo	MODgen
Synopsys FPGA Compiler	DesignWare
Synplicity	Custom algorithms
Viewsynth	Proprietary Interface

To illustrate the mechanisms and improvements using inference of ACTgen's datapath macros, consider the code in Figure 1. This function requires a magnitude comparator, a multiplexer to select which value to assign to output d and two adders as shown in Figure 1. The synthesis tool can synthesize the entire logic or it can infer datapath functions by the operators used in the code as indicated by the arrows. When several performance/cost variations are available, the synthesis automatically picks the one that best fits the defined design goals.

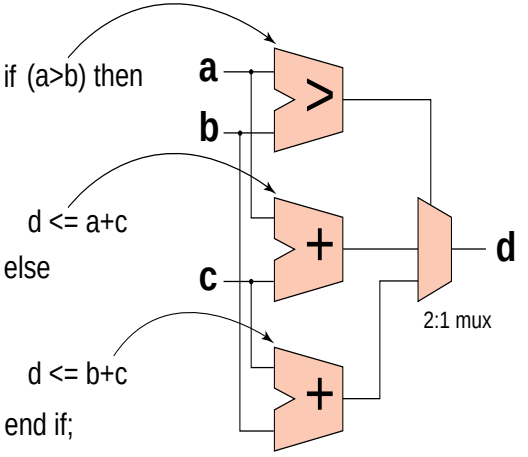

Figure 1 • *Automatic Inferencing Example.*

Functions built by ACTgen typically yield significantly improved results when compared to generic synthesis. A comparison of results are shown in Table 3.

These mecahanisms preserve the goal of vendor independence and enhances productivity by delivering improved results.

Table 3 • *Comparing Synthesis With and Without ACTgen*

	Generic Synthesis Only		ACTgen Usage		Percent Improvement	
Width	Area	Delay	Area	Delay	Area	Delay
8	131	18.2	76	16.6	42	9
16	286	22.8	196	19.2	31	16
24	450	28.0	289	23.6	36	16
32	626	28.3	414	24.0	34	15

Note: *Delays are in nanoseconds for an ACT 3, -3 speed device using pre-layout estimates.*

Some macro types, like memory, cannot be inferred directly and must be instantiated directly in the VHDL or Verilog code. This violates the goal of vendor independence; however, it is currently a limitation of most synthesis compilers. When instantiation is required, datapath macros are automatically generated by interactively accessing the ACTgen Macro Builer that is described in the following section.

ACTgen Macro Builder for Schematic Capture and Synthesis

The ACTgen Macro Builder provides a graphical user interface (GUI) for the Actel Datapath Synthesis engine. The ACTgen output can be logic symbols to support schematic capture or structural VHDL/Verilog to support synthesis instantiation. The initial GUI screen, as shown in Figure 2, provides "push button" selection of macros. Once selected, the macro can be configured on a subsequent GUI, as shown in Figure 3, for variation type, bit width, and required control signals. When the macro is configured, ACTgen reports performance and logic costs. If a designer is working with VHDL or Verilog, the ACTgen output is a structural netlist. When working with schematic capture systems, ACTgen outputs a symbol similar to the one shown in Figure 4 for the parameters defined in Figure 3. The macro can then be used just like any soft macro in the schematic capture environment including functional and back-annotated simulation. ACTgen supports CAE tools such as Cadence, Mentor Graphics, and Viewlogic.

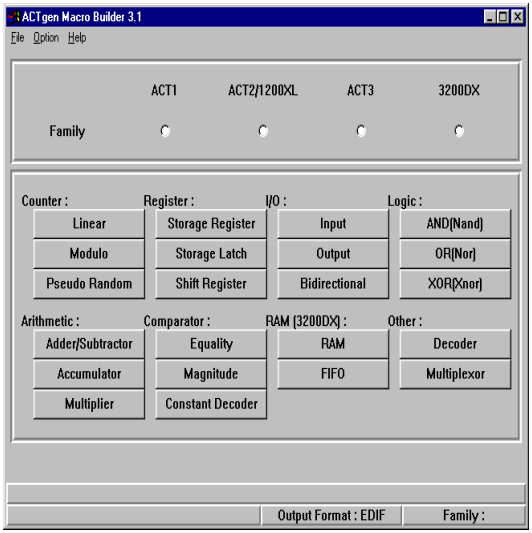

Figure 2 • *ACTgen Macro Type Selection.*

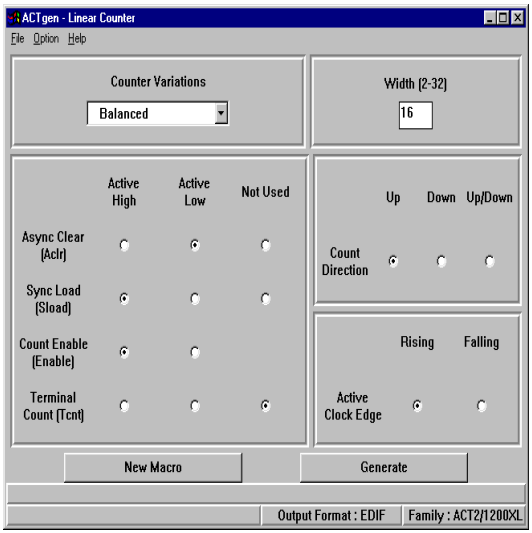

Figure 3 • *ACTgen Macro Configuration Screen.*

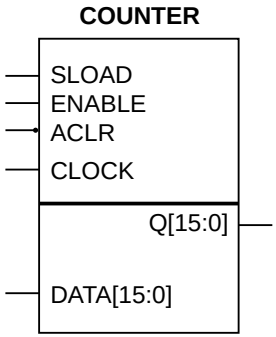

Figure 4 • *Symbol Generated from Figure 3.*

The ACTgen Macro Builder can also be used during early design prototyping regardless of design entry styles. In most cases, the datapath portion of the design will represent critical performance or logic resource bottlenecks. ACTgen rapidly builds and reports timing and the area for each type of macro, allowing the designer to make effective design tradeoffs early in the design cycle. For example, consider the function described in Figure 1. The same logic function can be described using one or two adders as illustrated in Figure 5. Using ACTgen, relative costs and performance for the two methods can be quickly evaluated preventing costly modifications late in the design cycle. Each approach has its own value. Option 1 provides the best performance with a high area impact; option 2 provides the best area with a slight decrease in overall performance.

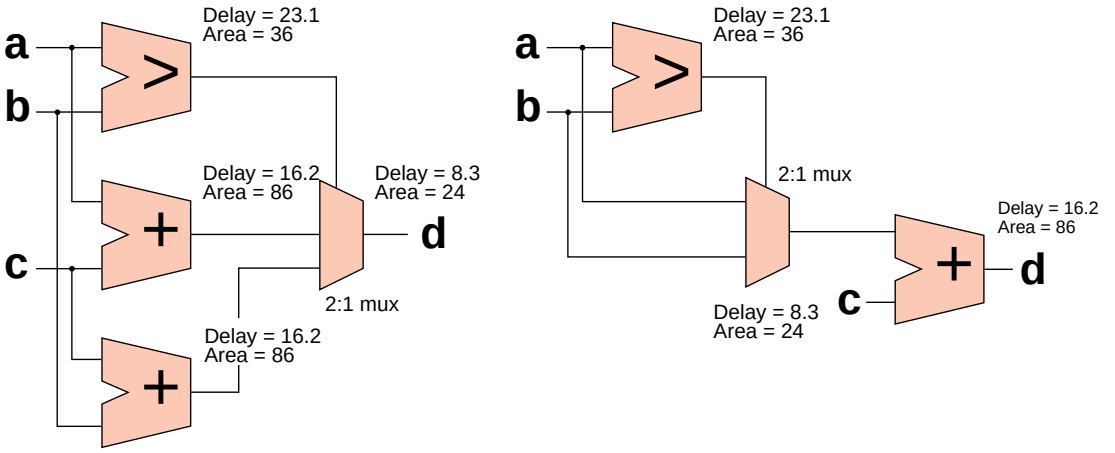

Figure 5 • *Implementation Options from Example of Figure 1*

Conclusion

Datapath functions have become the dominant logic type in complex logic devices. Consequently, overall quality of results is dependent on access to efficiently built datapath macros — Actel delivers with ACT gen. ACTgen can be accessed transparently by synthesis tools or can be used interactively by a designer. ACTgen provides both a mechanism to interface to schematic capture systems and provides an interactive platform for performance/cost trade-offs early in the design cycle.

Actel Corporation
955 East Arques Avenue
Sunnyvale, CA 94086
Tel: (408) 739-1010
Fax: (408) 739-1540

Actel Europe Ltd.
Daneshill House, Lutyens Close
Basingstoke, Hampshire RG24 8AG
United Kingdom
Tel: (+44) (1256) 305600
Fax: (+44) (1256) 355420

5192633-0